Stereo vision obstacle avoidance using depth and elevation maps

Taihú Pire, Pablo de Cristóforis, Matías Nitsche, Julio Jacobo Berlles
Computer Sciences Department
FCEyN - University of Buenos Aires.
tpire@dc.uba.ar dcristoforis@dc.uba.ar mnitsche@dc.uba.ar jacobo@dc.uba.ar

Abstract— **In this work an obstacle avoidance method is proposed for mobile robot autonomous navigation using a stereo camera as unique sensor. For each point of the world the disparity is computed from left and right images. With this approach, it is possible to determine the distance of nearby objects. Moreover, an elevation map is also computed for each pair of images. This map is used to remove the floor information in order to consider only the distances to the objects. Finally, using the depth and elevation maps an heuristic is defined to guide the robot avoiding obstacles. The method was successfully tested on indoor and outdoor environments.**

I. Introduction

The task of obstacle avoidance is one of the most important and researched problems in the field of mobile robotics. In order to navigate in the real world, it is necessary to detect those areas of the world that are dangerous or impossible to traverse. To accomplish the task of obstacle avoidance, a robot needs to know the distance of the objects around it. The most popular method to extract depth information from visual images is stereo vision which produces depth maps, the drawback of this technology is it requires a power platform processor.

There are several works applying stereo vision for obstacle avoidance on mobile robots. In [1] three methods of obstacle avoidance that use disparity maps are compared. The first phase is common to the three methods: all of them divide the disparity map in three parts or windows: left, central and right. The difference between methods resides in how they process the information from this regions. The first method proposes to compute the disparity average on each window. Then, the robot is guided to the area with the smallest value (assuming that the window has the less number of obstacles). The second method is an improvement of the first. In the beginning, it is only considered the central window of disparity map. The percentage of pixels with high disparity (higher than a given threshold) is computed. If the percentage is high, it is considered that in front of the camera there is an obstacle, and thus an avoidance action is performed. To choose a direction where the robot can move, side windows are analysed. The window with smaller disparity average is chosen. The third method takes into account the condition that the three windows have a high disparity average. If the robot falls into this case, it should perform an escape action, for example, to turn 180 degrees, or going backwards.

In [2], the use of just one part of the image it is suggested, a smaller area in compared to the whole image. This area is known as the *region of interest*. This approach reduces the time required to obtain the result. However, this method presents a drawback, given that if the region of interest is too small, relevant information of the scene could be ignored.

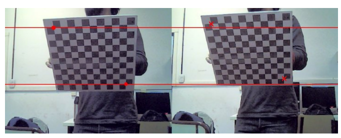
In the current work, a new obstacle avoidance method based on depth and elevation maps is proposed. The methods in [1] and [2] are taking as starting points.

II. Methods

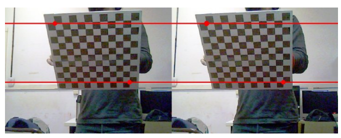
In this section we describe a method for the obstacle avoidance task, which consists of three main steps. In the first step the depth map is computed. The second step consists in the computation of the elevation maps and the third step correspond to the control algorithm which is based in the avoidance heuristic, that is, the manner by which the disparity map is analysed in each time step in order to command the robot.

A. Depth maps

To get depth maps it is necessary to have images in such a way that the corresponding pixels (homologous pixels) between one image and the other have the same height. For this reason, the images must be first rectified and vertical aligned as shown in Fig. 1. The Fig. 2 shows the resulting projective geometry after the images preprocessing. To obtained depth information of a particle real-world point its disparity is used. *Disparity* is the distance of a pixel in one image with its homologous in the other. *disparity map* is defined as the matrix which contains these values. The pixels belonging to nearby objects, have higher disparity than those belonging to far away objects. Homologous pixels are found using a matching algorithm.

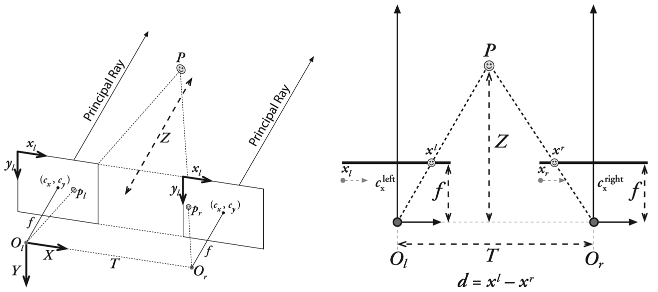


(a) Original Images captured during camera calibration.



(b) Images after rectification and vertical aligned.

Fig. 1. Images preprocessing just before the computation of disparity maps.



(a) (b)

Fig. 2. Rectified and vertical aligned images. The Z depth can be computed using similar triangles property. Images adapted from [3].

Let x^l y x^r the horizontals positions of homologous points in the left and right images, respectively. Therefore, the disparity d is given by $d = x^l - x^r$. Using similar triangles (1) it is possible to estimate the depth Z of one point in the real world given the focal distance f, the baseline T and disparity value d. The Fig. 2(b) shows the use of similar triangles geometry.

$$\frac{T-(x^l-x^r)}{Z-f} = \frac{T}{Z} \rightarrow Z = \frac{fT}{x^l-x^r} \rightarrow Z = \frac{fT}{d} \tag{1}$$

The real-world distance from one point to the camera is inversely proportional to disparity value (Fig. 3). If the disparity value is close to zero then small variations in the disparity generate big variations in depth. On the other hand, if disparity value is big, then small variations in depth do not generate depth differences.

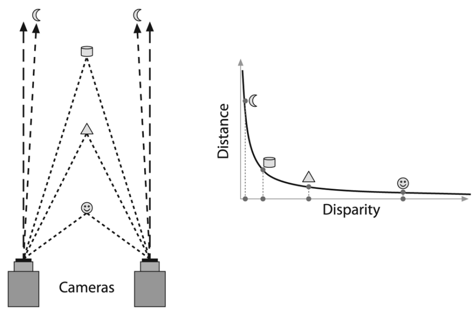
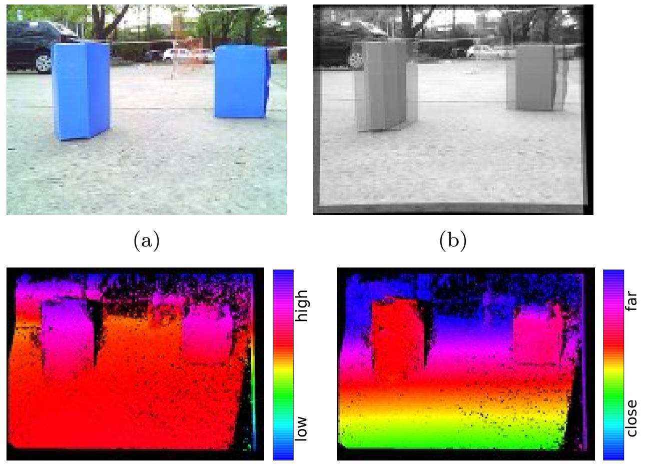


Fig. 3. Depth is inversely proportional to disparity, thus measurements are limited to nearby objects. Images adapted from [3].

Fig. 4 resumes the process to obtain disparity map given two images captured by a stereo camera. After rectification and vertical alignment of images, horizontal alignment is perform in order to define which element of the image will have zero disparity. This alignment can change according to the environment. In general, one takes as reference the farthest point of the environment. Finally, the disparity map is computed. HSV color-space is used to represent disparities values (see Fig. 4(d)), in order to facilitate visualization of objects closeness.

B. Elevation maps

An usual problem with obstacle avoidance methods based on disparity maps, is that the floor where the robot is moving, is frequently considered as a nearby object (depending of brightness, material, texture, etc) as shown Fig. 4(c).



(c) (d)

Fig. 4. Pictures where are shown partials results during the process to obtain disparity maps. (a) left original capture. (b) images horizontally aligned, one on top of another, in gray scale. (c) elevation map. (d) resulting disparity map.

To improve the avoidance heuristic, height information is added. For each pair of images, an elevation map is computed. This map is used to ignore the floor, in contrast to the approach of [2] which implements a horizontal line to limit the region of interest.

To compute the elevation map, the perspective transformation matrix Q is used to compute the 3D point representation for each pixel (x, y):

$$[X\,Y\,Z\,W]^\top = Q \times [x\,y\,d\,1]^\top$$
$$3dPoint(x,y) = (X/W, Y/W, Z/W)$$
$$\text{where } d \text{ is the disparity of the pixel } (x,y)$$

In this way, we can obtain a *3D Point Cloud* corresponding to the current robot view. To obtain the elevation map, the y coordinate of the 3D points (x, y, z) is saved. Observe that y is the height of the point in the world. Using this information it is possible to discriminate between points belonging to the floor and those which do not. Therefore, if a point's height is smaller than a given threshold it is considered as a *floor point*. The points which do not belong to the floor make up the region of interest.

C. Obstacle avoidance

The obstacle avoidance control algorithm is based on the *Threshold Estimation Method* [1]. This method uses disparity maps to approximate the distance of objects to the robot. In this work a variation of this algorithm is developed.

First, the disparity map is divided in three vertical windows of variable size. The central window was configured bigger than the side windows. In this way, we attempt to minimize the limitations caused by camera vision angle and try to cover an area bigger enough to detect objects localized in front of the camera as potential obstacles. Moreover, the side windows were configured with different size in order to reduce the noise generated by the lack of information in the border of the disparity map (black areas in the borders of Fig. 4(d)).

The second step consists in the estimation of the percentage of disparity map pixels which represent nearby objects. These percentages are obtained for each window. The pixels whose disparity are greater than a given threshold are considered nearby (while smaller is the threshold, greater will be the mount of pixels considered as nearby). Also, the pixels which do not have disparity information are taking into account, given that the disparity maps present in general a great number of areas where it is not possible to find matching pixels between images. To avoid that these pixels take precedence over the high disparity pixels, they are taking into account weighting at 30% of high disparity pixel value. It should be pointed out that, for performance efficiency, the side areas are not processed until it is really necessary, as it is shown bellow.

The next step is to take the movement decision. First, only the central area is analysed. If the percentage of high disparity pixels is smaller than a given threshold (to instance 30%) it is considered that the road is free of obstacles and the robot can continue its trajectory. In contrast, if the threshold is exceeded, the percentages of side windows are compared. The window with the smallest value indicate the direction with few or no obstacles, therefore the robot is turned accordingly.

III. Results

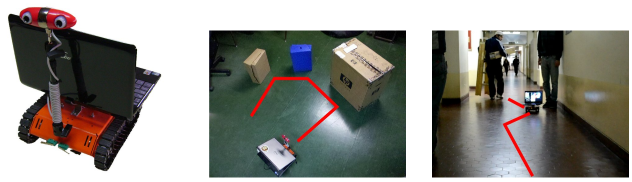
To carry out the experiments, the *ExaBot* robot [4] was used. It was designed and manufactured in the FCEyN - UBA Argentina. It is a differential mobile vehicle with two caterpillar tracks and two DC motors, which allow to perform 2-DoF egomotion on the plane. For the experiments, the robot was configured with a stereo camera and a laptop.

The stereo vision equipped on the robot is a *Minoru* camera. The camera's specifications are: 60 mm of base stereo, 2.0 USB connection, from 320×240 to 1280×480 output resolution sizes (640×480 is used), $15 \sim 30 fps$ and 1.5 W power consuming. The laptop used for the experiments has an Intel Core $i5 \sim 2.67$ GHz processor and 4 GB of RAM. During the experiments, a processing of $8 \sim 10$ frames per second was achieved.

The method was tested on indoor environments with objects of minimum size of 40 $cm \times 30$ $cm \times 15$ cm. In Fig. 5(b) and Fig. 5(c), images of the robot executing the obstacle avoidance algorithm are presented. Some preliminary experiments were also conducted in outdoor environments (Fig. 4).

Table I shows the time (in milliseconds) consumed by each method step. It can be seen that the largest amount of time corresponds to the disparity map computation. Disparity maps is computed using *LIBELAS* library [5].

At this moment obstacle avoidance implementation uses only a horizontal line to limit the region of interest by eliminating the floor. The use of elevation maps has been proposed and implemented but not used in the proposed method yet. Anyway, elevation maps were computed during the experiments and the floor was correctly identified as such using this technique.



(a) (b) (c)

Fig. 5. (a) Exabot robot mounted with a Minoru stereo camera and standard laptop. (b) and (c) trajectories performed by the robot during obstacle avoidance experiments. In `http://youtu.be/Gxwl0E4BhbQ` is possible to watch one of the experiment videos.

Method step	time in ms
Disparity map	78
Elevation map	10.7
Control	1.3
Total	90

Table I

Frame processing time (in milliseconds) who takes each method phase.

IV. Discussion

During the experiments, we could observe that the quality of object detection depends directly on the size, texture and brightness. On the other hand, we observe the high computing cost of disparity maps. To overcome this issue, a possible optimisation is to compute the disparity maps corresponding to each window of image on demand, i.e. compute the side disparity maps when it is not possible to move forward.

V. Conclusion

In this work, an obstacle avoidance method based on depth and elevation maps was presented. In the experiments, the method uses a horizontal line, instead of height information, to ignore the floor. However, elevation maps were computed during algorithm execution and the floor was correctly identified. The method was successfully tested on indoor and outdoor environments.

References

[1] Ioannis Kostavelis, Lazaros Nalpantidis, and Antonios Gasteratos. Comparative presentation of real-time obstacle avoidance algorithms using solely stereo vision. In *IARP/EURON International Workshop on Robotics for risky interventions and Environmental Surveillance-Maintenance*, Sheffield, UK, January 2010.

[2] Anwar Hasni Abu Hasan, Rostam Affendi Hamzah, and Mohd Haffiz Johar. Region of interest in disparity mapping for navigation of stereo vision autonomous guided vehicle. In *Proceedings of the 2009 International Conference on Computer Technology and Development - Volume 01*, ICCTD '09, pages 98–102, Washington, DC, USA, 2009. IEEE Computer Society.

[3] Gary Bradski and Adrian Kaehler. *Learning OpenCV, Computer Vision with OpenCV Library*. O'Reilly Media, first edition, 2008.

[4] Pablo de Cristóforis, Sol Pedre, Andrés Stoliar, and Javier Caccavelli. Exabot: a mini robot for research, education and popularization of science. *VI Latin American Summer School in Computational Intelligence and Robotics - EVIC2009*, december 2009.

[5] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision*, Queenstown, New Zealand, November 2010.